BPM and Web Dynpro for Java Integration – Guidelines

June, 2009

Team Members:

Sunil Nair
SAP NetWeaver RIG APJ

Makoto Sugishita
SAP NetWeaver RIG APJ
Disclaimer

This presentation outlines our general product direction and should not be relied on in making a purchase decision. This presentation is not subject to your license agreement or any other agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or to develop or release any functionality mentioned in this presentation. This presentation and SAP's strategy and possible future developments are subject to change and may be changed by SAP at any time for any reason without notice. This document is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP assumes no responsibility for errors or omissions in this document, except if such damages were caused by SAP intentionally or grossly negligent.
Content

Overview – Business Process Management

- Model Based Business Process Management
- Composite Business Process
- Business Process Modeling Notation (BPMN)
- User Interface and Human Activity
- Business Process Desk
- Task Execution User Interface

Overview – Web Dynpro for Java

- At a glance
- Web Dynpro Tools and Code Generation
- Web Dynpro DC
- Web Dynpro Component and Componentization
- Context Mapping and UI Element Binding

Integrate Web Dynpro Java with Business Process Management

- How to embed WebDynpro into Business Process Management – Step by Step

Guidelines for Web Dynpro Development

- Recommendations, best practices, dos & donts
- Important SAP Notes and links

© SAP 2009 / SAP NW RIG APJ Page 3
Model-Based Business Process Management

SAP NetWeaver 7.1 EHP1

Composition Environment 7.1 EHP1

Process Integration 7.1 EHP1

SAP Application Core Processes

Business Object

Business Object

Human-Centric
- Executable models
- Process collaboration
- Human interaction management

Packaged Processes
- Descriptive models
- Embedded workflow
- Expose services / events

System-Centric
- Executable models
- Process automation
- Service and event infrastructure

B2B

Non SAP
Composite Business Processes

Characteristics

- **Graphical modeler based on BPMN**
  - Eclipse-based
  - Support key process concepts (workflow, event, task, context, roles, UI) throughout their lifecycle
  - Combine human interaction and system integration in one model

- **Direct path from business view to process execution**
  - Nothing ‘lost in translation’
  - Dynamic role-based views
  - Single active model
  - Flexible execution

- **Embedded in SAP NetWeaver CE**
  - Integrated composition experience
  - Common UI technology
  - Service-based connectivity
BPMN – Sequence Flow Overview

**Activity**
- A step in the process
- Represents work or action performed

**Gateway**
- Controls flow branching, merging and parallel actions
- Pure logic – does not do the decision itself

**Event**
- A signal that „something has happened“
- Can start, pause and resume or interrupt and redirect a process or activity
BPMN – Type of Activities

**Human Activity**
- Activity that has to be processed by a natural Person

**Automated Activity**
- Activity that is processed by a system

**Sub-process**
- A compound (decomposable) Activity that holds a sub-process modeled with BPMN – can be shown collapsed or expanded
BPM Process Desk

Create Purchase Request

- Status Information
- Delegating or revoking task
- Attaching notes or files to the active task or the whole process
- Process view for showing process runtime information
The Process Desk is the interaction layer between the user and process.

Process Tasks assigned to a user show up in the UWL of the Portal.

Tasks are displayed in the Task Execution User Interface.

PurchaseRequest

- **Started at**: 12/14/06 4:48:37 PM
- **Due at**: Medium
- **Status**: In Progress
- **Owner**: Demo, Dee
- **Priority**: Medium
- **Process**: Investment Approval Process

Enter Purchase Request

- **Requester**: Demo, Dee
  - **Country**: United States
- Tasks are displayed in the task viewer User Interface
- Embeds the Web Dynpro assigned to the task at design time
**Task Execution User Interface**

**Attachments and Notes**

- **Can be added to tasks only at runtime**

*PurchaseRequest*

<table>
<thead>
<tr>
<th>Started at</th>
<th>Due at</th>
<th>Status</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/14/08 4:48:37 PM</td>
<td>---</td>
<td>---</td>
<td>Investment Approval Process</td>
</tr>
</tbody>
</table>

**Owner** Demo, Dee  
**Priority** Medium

- **Viewable in proceeding tasks of the same process instance**

**Notes**

**Attachments**

- No Attachments

© SAP 2009 / SAP NW RIG APJ Page 11
Task Execution User Interface
Process Template

- Process Template can be displayed

- Current user sees current steps within the process
Content

Overview – Business Process Management
- Model Based Business Process Management
- Composite Business Process
- Business Process Modeling Notation (BPMN)
- User Interface and Human Activity
- Business Process Desk
- Task Execution User Interface

Overview – WebDynpro for Java
- At a glance
- Web Dynpro Tools and Code Generation
- Web Dynpro DC
- Web Dynpro Component and Componentization
- Context Mapping and UI Element Binding

Integrate WebDynpro Java with Business Process Management
- How to embed WebDynpro into Business Process Management – Step by Step

Guidelines for WebDynpro Development
- Recommendations, best practices, dos & donts
- Important SAP Notes and links
At a glance

- **SAP NetWeaver UI Technology for developing data-driven web applications**
- **Use of declarative and graphical tools reduces the implementation effort**
- **Offers the following advantages for application developers:**
  - Enforces separation of business logic and display logic
  - Declarative UI development independent of client technology (like web browser, smart client, mobile device)
  - Uniform metamodel for all types of user interfaces
  - Comprehensive library with standard, complex and graphical UI elements
  - Connectivity to backend systems using, for example, web services, Enterprise JavaBeans, and adaptive RFC
  - Accessibility support
  - Adobe Forms integration
  - The pages created by WebDynpro can be personalized, user or roles based
  - You have the choice to work with visual models or code editors
WebDynpro Tools and Code Generation

- **A glance at Web Dynpro Tools**
  - Data Modeler: Sharing data between classes within a component
  - Navigation Modeler: Creation of navigation links between different UI views
  - View Designer: Define the layout and data sources of UI elements
  - Controller Editor: Define data storage, instance methods, events, navigation plugs and actions
  - Message Editor: Message and text string definition
  - Model Importer: Wizards to create various types of model object

- **Support design and development process for UIs and entire applications**
- **Use of declarative and graphical tools reduces the implementation effort**
- **Reduces coding efforts by generating entities based on defined meta-data**
  - Source Code Generation for Controller Classes and Interfaces
  - Classes contain user coding areas where additional code can be implemented
  - Interfaces for context objects (typed nodes and node elements)
  - Interfaces for message texts
  - Model classes
Web Dynpro Development Component (DC)

- Development and build unit

- Basic reusable units of an application
  - One DC can use another DC by referring to the public interfaces (public parts)

- Web DC comprises of
  - Web Dynpro-specific developments, such as Web Dynpro components, models, component interfaces, views, and so on
  - Provides storage buffer for DC metadata which serves to define application relationships between different DCs

- Relation between Web Dynpro DC and Web Dynpro Component
Web Dynpro Component and Componentization

Overview

- Fundamental building block from which all applications are constructed
- Basic unit of development and reuse
- Atleast one Web Dynpro Component in a Web Dynpro application
  - Applications used to implement real-life business scenarios are frequently constructed from a hierarchy of multiple components
- An aggregation of independent, yet interrelated Java classes

General Architecture of Web Dynpro Component

- **Component Controller**: Master Controller generated automatically
- **Custom Controller**: Similar but created by explicit design time declaration
- **Component Interface**: Defines set of publicly accessible entry points
- **Window**: Atleast one view and default interface with which user interacts
- **View**: Visual component inside a window with which a user interacts
- **Model**: Provides access to functionality such as BAPI calls or Web services
- **Application**: Defines an entry point into a Web Dynpro component
Focus on implementing the functionality using reusable Web Dynpro components

Focus on reusable Web Dynpro components, not on the web pages

Component Hierarchy

Root Component
- Deliver functionality of entire Web Dynpro business application, and can’t be reused individually.
- Manage creation, destruction and interaction of child component instances
- Entry point into business functionality and cannot be faceless

Intermediate Component
- Performs specific role within the scope of Web Dynpro application
- Designed to function within scope of single business application and hence not suitable for reuse

Utility and Model Component
- Utility Component : Small and frequently used units of functionality
- Provides access to functionality such as BAPI calls or Web services
Share data between controllers

Data Storage and consistency

Putting data on the screen

Data binding enables interaction between UI elements and context elements

Changes of UI element are transferred to context and vice versa
Data passes from the model object to the screen using this chain of connections:

- The context of the component controller is bound to the model object
- The view controller context is mapped to the component controller context
- The UI elements are bound to the view controller’s context
Content

Overview – Business Process Management

- Model Based Business Process Management
- Composite Business Process
- Business Process Modeling Notation (BPMN)
- User Interface and Human Activity
- Business Process Desk
- Task Execution User Interface

Overview – Web Dynpro for Java

- At a glance
- Web Dynpro Tools and Code Generation
- Web Dynpro DC
- Web Dynpro Component and Componentization
- Context Mapping and UI Element Binding

Integrate Web Dynpro Java with Business Process Management

- How to embed WebDynpro into Business Process Management – Step by Step

Guidelines for Web Dynpro Development

- Recommendations, best practices, dos & donts
- Important SAP Notes and links
Integrate Web Dynpro Java with BPM
Overview

- Define Complete Event in Component Controller and expose it to the Interface Controller
- Expose needed context structure from Component Controller to the Interface Controller (only context structure needed by BPM process must be exposed)
- Fire Complete Event in the Component Controller (indicates BPM, that UI task is completed by end user)
- Add Web Dynpro Component to Public Part and set Dependency
- Modify BPM Project
  - Create BPM Task and assign UI to that
  - Map WebDynpro Context to Data Object
Web Dynpro Component Controller

- Define the following Event(s) in Component Controller. You can use arbitrary name for these events.
  - Complete
  - Error (optional)

Web Dynpro Local Component Interface Controller

- Create the events within Web Dynpro Local Component Interface for communicating with SAP NetWeaver BPM by pasting the Component Controller Event(s).

You can Copy Paste Events from Component Controller to Interface Controller.
Expose Context Structure to Interface Controller

Web Dynpro Component Controller
- Use only “built-in” data types for context attributes

Web Dynpro Local Component Interface Controller
- Only context nodes defined at design time are visible to SAP NetWeaver BPM

You can Copy Paste Context Structure from Component Controller to Interface Controller
Web Dynpro Component Controller

- Create methods within Web Dynpro Component Controller for firing “Complete” and/or “Error” events
- Then bind this method for example to an ActionButton

```java
public void wdDOLocationStateChange()
{
    // Method declared by application.
}
```

```java
public void complete()
{
    // Method declared by application.
    wdThis.wdFireEventComplete();
}
```
Add Web Dynpro Component to Public Part and Set Dependency

Development Infrastructure perspective
Set dependency between Development Component of SAP NetWeaver BPM and Web Dynpro

Web Dynpro Component Controller
- Add to Public Part
Modify BPM project
Create Task for Human Activity and assign UI

- Create a task within SAP NetWeaver BPM

- Maintain Tasks with additional Information about roles and behavior
  - Import Web Dynpro Component and assign User Interface to Task
Modify BPM project
Assign Task to Process Model and Map data

- Assign Task in property window of human activity

- Web Dynpro Contexts are shown in the mapping
Appendix: Runtime Behavior
Completion Event and Error Event

By triggering the Event for “Completion”,

- The human activity completes
- Dialog pops up asking whether to close the window of the BPM task.

By triggering the Event for “Error”,

- The process became the error status (You can change the status back in NWA)
- Error message appears
Content

Overview – Business Process Management
- Model Based Business Process Management
- Composite Business Process
- Business Process Modeling Notation (BPMN)
- User Interface and Human Activity
- Business Process Desk
- Task Execution User Interface

Overview – Web Dynpro for Java
- At a glance
- Web Dynpro Tools and Code Generation
- Web Dynpro DC
- Web Dynpro Component and Componentization
- Context Mapping and UI Element Binding

Integrate Web Dynpro Java with Business Process Management
- How to embed WebDynpro into Business Process Management – Step by Step

Guidelines for Web Dynpro Development
- Recommendations, best practices, dos & don’ts
- Important SAP Notes and links
Recommendations for storing business data

- Volume of data low and process is small, you may want to pass and store the entire data into BPM Context. Nevertheless, BPM has no restrictions on data volume.
- It is recommended that you pass only the key to the BPM and get updated data

  - BPM stores instance specific data only till the process is alive and the data is needed
  - For processes that are very long and the data is expected to change frequently during the process execution, then pass the key and fetch the updated information
  - External data store is if user wants to save the temporary data
  - Take into consideration the supported and unsupported data types (see next slide)
Guidelines for Web Dynpro Development
Recommendations, best practices, dos & don'ts

Supported Data Types

- Built-In SimpleTypes: binary, boolean, byte, date, decimal, double, float, integer, long, short, string, time, time stamp

Non Supported Data Types

- Dictionary types (predefined types, for example, currency), core data types (for example, amount, duration, attachment), Java native types

Limitations with CE7.1.1

- Reverse Context Mapping
  - Web Dynpro component’s interface controller context nodes have the property Input Element set to true. This property must be set to false.

- Model Binding
  - Web Dynpro component’s interface controller context nodes, which are bound to Web Dynpro model (CMI) are instantiated after the point in time when BPM can access Web Dynpro context. Due to this the use of Web Dynpro context nodes, which are bound to used model are not supported.

- Dynamic Context Structure Modification
  - For creation of runtime data, mappings and rules, BPM takes Web Dynpro component’s structure from introspection of component at design time
Guidelines for Web Dynpro Development
Recommendations, best practices, dos & don'ts

Recommendation on Web Dynpro UI Design

- UI for each task should be user friendly. The end user should be able to understand what user actions are necessary to complete the task, otherwise the user might move to other operation without completing the task
- Display all necessary information that is required to complete the task on the screen
- Avoid UI Transitions that confuse the end users. However, UI transition in a Task is sometimes necessary because putting consecutive human task works differently as the modeler expected (see below)
- It’s better to validate data in UI screen if possible, in order to avoid the redundant human work, e.g. reject
- Reusability should be considered

- If you model several consecutive human tasks, the user has to complete the first task, refresh UWL and open the second task from inbox again, even when the same user processes those tasks
- The behavior of NW BPM is not identical as that of Guided Procedure
Guidelines for Web Dynpro Development
Recommendations, best practices, dos & don'ts

Recommendation on DC and WD Component Design

- Developer's choice or as per organizational development guidelines
- One Process DC per process with WD DC having one WD Component for each task or human activity. Additionally, can have WD Utility DC whenever needed
- Reuse tasks, UIs, Interfaces etc. Whole SOA is based on reusability
  - Request and Approve/Reject Screens can be reused with true/false flags

Recommendation on Process Design

- No restrictions as long as the end user is comfortable
- Typically, medium size processes can have 6-7 Human Activities i.e. UI screens
- Very Long processes can have 10 – 15 Human Activities
- Better to cut the processes into smaller chunks, better reusability and maintenance
Guidelines for Web Dynpro Development

Important SAP Notes and links

**SAP Notes**

Note 1263355 - Task UI Integration Limitations

Note 1266539 - XSD-related limitations for SAP NetWeaver

**Links**

**SAP NetWeaver CE7.1 Enhancement Package 1 Library – Modeling Processes**
(http://help.sap.com/saphelp_nwce711/helpdata/en/ce/19dc55105b46a0b498af9d840a93a8/frameset.htm)

**SAP NetWeaver BPM Troubleshooting Guide**

**SAP NetWeaver BPM : Flash Island Meets BPM**
(https://www.sdn.sap.com/irj/scn/go/portal/prtroot/docs/library/uuid/9006a1ee-7410-2c10-66a0-e6214bba38e4)

**SAP NetWeaver BPM Resource Center**
(https://www.sdn.sap.com/irj/irj/sdn/nw-bpm-info)

**SAP Business Process Management on SDN**
(https://www.sdn.sap.com/irj/irj/sdn/nw-bpm)
THANK YOU FOR YOUR ATTENTION!

QUESTIONS  –  SUGGESTIONS  –  DISCUSSION